

**REMARKS**

Claims 1-26 are pending.

**Rejections Under 35 U.S.C. §103(a)**

Claims 1-4, 10-13 and 26 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent App. Pub. No. 2002/0091990 ("Little") in view of U.S. Patent No. 5,987,247 (Lau). Claims 5-9 and 14-25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Little and Lau and further in view of U.S. Patent No. 6,223,343 ("Hopwood"). Reconsideration is respectfully requested.

**The Present Invention**

As explained in previous responses, the present invention is directed to an approach to software development that starts at a much higher level of business process abstraction than traditional approaches to business software development, such as the Rationale Rose-based system described in Little. Indeed, as described in paragraphs 0010 and 0011 of the Background of the Invention section of the instant application, while the use of "use cases" and "object models" in systems like that described in Little provides "a useful abstraction [that] allow[s] the software developer to create software with a view toward specific situations that the software will be expected to handle," such "use cases still have the drawback of being, in many situations, at a much lower level of abstraction than the requirements for which the software is designed." What the present invention does differently is to begin the process of software development at a much higher level of abstraction.

As described in paragraph 0031 of the instant specification, a company can be defined by the set of processes that take place within it. For example, an airline performs a process of receiving a reservation over the Internet, as well as a process of receiving luggage at a check-in counter and transporting it to the appropriate plane. Some of the processes, such as the receipt of a reservation over the Internet, may benefit from a high degree of automation by business software. Other processes, such as moving luggage from the check-in counter to the airplane, cannot easily be automated by business software, because, for example, they must

be performed manually. Nevertheless, it is helpful to define all the processes of a company and the interrelationships between them.

According to the present invention, “blueprints” are used to define the business processes of a company and the interrelationships between them. As expressly defined in the instant specification a “blueprint” is “a collection of documents, called artifacts, that can be used to create a cross-referenced representation of the business processes that occur within an enterprise (Spec, ¶ 0034).” As expressly recited in independent claims 1, 15, 25 and 26, the claimed “blueprints” comprise information relating to a particular industry and provide “a cross-referenced representation of business processes that occur within the enterprise.” Once such “blueprints” have been created, certain business processes may be selected for automation using a business software solution. The artifacts for the business processes to be automated can be used as a guide for a programmer to create the business software solution. And because the blueprints describe the interrelationships between business processes, a change to a business process can be propagated throughout the business software solution by following the cross-references to other processes that are provided in the artifacts of the changed process. These features are not disclosed in Little.

#### **Neither Little nor Lau Teach or Suggest “Blueprints” As Claimed**

The Examiner acknowledges that “Little doesn’t explicitly disclose wherein it [*i.e.*, a “blueprint”] provides a cross-referenced representation of business processes that occur within the enterprise” (Office Action, p. 3). But the Examiner asserts that “Lau in an analogous art and similar configuration discloses in (13:65-14:20) that,

‘ ...artifacts are generated by the parsing and importing subsystem by parsing the interface definition stored in the source files ... Once the interface definitions have been parsed, the parsing and importing subsystem then stores (*i.e.*, imports) the parsed interfaced definitions... Once the parsed interface definition has been imported into the data model, the same template based code generation scheme is used by the code generator to emit the necessary code to make the generated object correspond to the Business Object environment...’”

and that, therefore, “it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine, Little and Lau, because it would enable referencing parsed definitions for corresponding Business object environments as suggested by Lau

above” (Office Action, p. 4). However, the applicants submit that the Examiner is misconstruing the teachings of Lau and that the cited portion of Lau does not teach or suggest the claimed concept of a “blueprint” that comprises information relating to a particular industry and that “*provides a cross-referenced representation of business processes that occur within the enterprise,*” as recited in the each of independent claims 1, 15, 25 and 26 (emphasis added).

Lau relates to object-oriented framework technology, and in particular, is directed to a system for “building a framework of objects corresponding to a design for an object oriented application” (col. 5:30-40). As described in Lau, a “Business Object refers to an object [in the object-oriented programming sense] that encapsulates business logic data or information ...” (col. 7:48-49). Essentially, they represent *things* in a business. “Examples of ...Business Objects include ‘person’ and ‘policy’ ... [and] ‘customer’” (col. 7:60 – 8:1).

The portion of Lau alleged to teach the “cross-referenced representation” feature of the claimed invention instead describes a “Parsing and Importing Subsystem” of the framework building system of Lau. In particular, the cited portion describes how “[t]he framework building system ... converts interface definitions that were written in interfaces used in conventional editors into Business Object corresponding classes” (col. 13:65-14:1). As further described, the process involves parsing the interface definitions that were written using conventional editors (*e.g.*, IDL, JAVA) and then storing those parsed interface definitions into the framework building system’s data model (col. 14:2-12). “Once the parsed interface definition has been imported into the data model, the same template based code generation scheme is used by the code generator to emit the necessary code to make the generated object correspond to the Business Object environment” (col. 14:16-20).

Thus, what the cited portion of Lau is describing is a process that converts interface definitions for objects that were written using conventional editors into a “generated object” that works like (*i.e.*, “corresponds to”) any other “Business Object” in Lau’s framework building system. That conversion process has nothing to do with the applicants’ concept of “blueprints” that provide “a cross-referenced representation of business processes that occur within the enterprise,” as claimed.

The Examiner’s assertion that the cited portion of Lau teaches “referencing parsed definitions for corresponding Business Object environments” simply mischaracterizes what

the cited portion of Lau teaches. As explained above, when Lau states that “[o]nce the parsed interface definition has been imported into the data model, the same template based code generation scheme is used by the code generator to emit the necessary code to make the generated object *correspond to the Business Object environment*,” (col. 14:16-20) what Lau means is that the interface definitions for objects that were written using conventional editors are converted into a “generated object” that works like any other “Business Object” in Lau’s framework building system. The reference to “Business Object environment” is not a reference to a business process that occurs within an enterprise, as the Examiner seems to imply, but rather is a reference to the overall structure and function of a “Business Object” in Lau’s framework building system. Moreover, also contrary to the Examiner’s assertion, there is no discussion of any “cross-referencing” in the cited portion.

For the foregoing reasons, the applicants respectfully submit that Lau does not teach or suggest the claimed concept of a “blueprint” that comprises information relating to a particular industry and that “*provides a cross-referenced representation of business processes that occur within the enterprise*,” as recited in the each of independent claims 1, 15, 25 and 26 (emphasis added). No combination of Lau with Little would produce the claimed invention. Therefore, the applicants respectfully submit that those claims patentably define over Little and Lau, alone or in combination.

**Little Also Does Not Teach or Suggest “Providing Documentation” As Claimed**

With respect to the claimed feature of “providing documentation” in independent claims 1 and 15 and “recording documentation” in claim 25, the Office Action continues to refer to the following statement in paragraph [0145] of Little: “A use case diagram provides a functional specification of a system and its major processes, and describes the problem that needs to be solved” (Office Action, p. 3). The Examiner appears to interpret this description as “providing documentation” (*Id.*)(emphasis added). But there is no explanation of how this statement teaches or suggests that the provided documentation “*specifies a relationship between at least two functional components, thereby enabling traceability between the at least two functional components*,” as recited in claim 1 or the similar recitations in claims 15 and 25 that the documentation “*specifies a traceable relationship between ... the one or more functional components*” (emphasis added). The applicants respectfully submit that the

**DOCKET NO.:** USYS-0203/BB012AUSC1  
**Application No.:** 10/731,846  
**Office Action Dated:** 04/02/2008

**PATENT**

Examiner has not given proper weight to the highlighted language of these claims. Because Little fails to teach or suggest this additional feature of independent claims 1, 15 and 25, the applicants submit that those claims patentably define over Little and Lau for this additional reason.

For all the foregoing reasons, the applicants respectfully submit that independent claims 1, 15, 25 and 26 patentably define over Little and Lau, whether alone or in combination with any of the other cited art of record. Hopwood does not cure the deficiencies of Little and Lau. Moreover, inasmuch as the remaining claims depend either directly or indirectly from one of those independent claims, the applicants submit that they too patentably define over the cited art of record.

**DOCKET NO.:** USYS-0203/BB012AUSC1  
**Application No.:** 10/731,846  
**Office Action Dated:** 04/02/2008

**PATENT**

### **CONCLUSION**

For all the foregoing reasons, the applicants respectfully submit that the present application is now in condition for allowance.

Date: August 4, 2008

/Steven B. Samuels/

Steven B. Samuels  
Registration No. 37,711

Woodcock Washburn LLP  
Cira Centre  
2929 Arch Street, 12th Floor  
Philadelphia, PA 19104-2891  
Telephone: (215) 568-3100  
Facsimile: (215) 568-3439

**Correspondence address:**  
**Richard J. Gregson, Esq.**  
**Unisys Corporation**  
**Unisys Way**  
**Blue Bell, Pennsylvania 19424-0001**